ANALISIS DESKRIPSI BAHASA YANG BERSESUAIAN DENGAN MODEL AUTOMATA

Sarwah[1]
SMAN 19 Luwu Utara[1]

Sunyi.lembah@ymail.com[1]

Bahasa formal merupakan satu-satunya bentuk bahasa yanga dapat dipakai berkomunikasi antara manusia (programmer komputer) dengan komputer. Karena bahasa ini disusun berdasarkan aturan yang ditetapkan yang disebut gramatika. Gramatika ini mempunyai bentuk produk yang berbeda-beda. Selanjutnya menghasilkan bahasa yang berbeda-beda jenisnya menurut gramtika tadi. Tiap jenis bahasa formal ini, akan dikenali oleh suatu peralatan pengenal yang disebut automata. Artinya hanya automata yang dapat mengartikan sebuah bahasa dalam bentuk output.

1. Pendahuluan

Komunikasi dengan komputer bisa terjadi karen ada bahasa formal yang memenuhi konsep aljabar dengan sifat; satu simbol/kata hanya mempunyai satu arti. Bahasa formal ini dibentuk berdasarkan aturan yang disebut ***gramatika*** (tata bahasa). Gramatika formal dibentuk dengan memisalkan V≠Ø yang unsurnya adalah simbol-simbol (selanjutnya disebut ***alfabet***) dengan aturan produksi tertentu dalam menghasilkan kalimat.

Berdasarkan pembentukan ini aturan produksi (production rules), bahasa formal terdiri dari beberapa type/jenis (dalam hal ini hanya akan dibahas tiga) yang masing-masing mempunyai model matematika yang berbeda. Salah satu bentuk produksi bahasa formal adalah (ξ,β) atau $\xi \rightarrow \beta$ dengan $|\xi| \leq |\beta|$ yang disebut bahasa tipe-1 yang biasa disebut dengan nama sensitif konteks. Sedangkan tipe bahasa lainnya adalah bahasa tipe-2 yang dinamai sebagai bahasa konteks bebas, bahasa reguler untuk bahasa tipt-3.

Bahasa-bahasa formal tadi dapat dikenal oleh suatu pengenal yang disebut automata. Tiap model deskripsi bahasa hanya bisa dikenal oleh satu automata yang sesuai bentuk produksinya. Berdasarkan hal tersebut diatas maka penulis akan mengungkapkannya dalam bentuk tulisan yang berjudul "Analisis Deskripsi Bahasa Yang Bersesuain dengan Model Automata".

Masalah struktur aljabar merupakan unsur penujangnya dalam hal ini pengertian dasar pembentukan monoid. Selain itu, juga diperlukan pengertian dasar dari graph.

Ide-ide dasar dalam bentuk gramatika dan bahasa sehari-hari akan dipaparkan untuk menunjukkan bahwa dalam suatu kalimat, aturan pembentukannya dapat digunakan untuk membentuk kalimat tertentu dalam gramatika dan bahasa formal.

Gramatika formal dibentuk dengan memisalkan suatu yang unsurnya adalah simbol-simbol (selanjutya disebut alfabet) dengan aturan produksi tertentu dalam menghasilkan suatu kalimat tertentu. Selanjutnya bahasa formal dibentuk atas gramatika tersebut.

Hal yang lebih penting dan ditekankan dalam tulisan ini adalah analisis kesesuaian bahasa dengan automata agar bisa saling mengenal dan menerima.

Sedangkan konsep automata sendiri tidak dibahas terlalu detail, seperti automata beroutput. Jadi hanya dibatasi pada pengetian dasar dari automata (khususnya automata berhingga), yaitu bagaimana kriteria automata yang bisa menerima bahasa tertentu. Oleh karena itu selain kriteria automata, juga diperlukan kriteria bahasa yang dapat diterima oleh automatanya.

2. Kesesuaian Bahasa (Formal) Dengan Automata

Pada bagian sebelumnya telah dijelaskan bagaimana proses pembentukan suatu bahasa (formal) yang dimulai dari simbol-simbol kemudian dari simbol-simbol tersebut dibentuk string. Selanjutnya string membentuk suatu bahasa melalui gramatika dengan memakai aturan produksi.

Automata

Automata berhingga merupakan salah satu jenis automata yang ditujukan sebagi mesin pengenal sekaligus penejemah suatu bahasa formal, biasa juga disebut mesin moore yang dibangun oleh bahasa reguler. Automata berhingga ini diprogramkan untuk menafsirkan kata-kata atau simbol-simbol yang terdapat pada string input. Automata berhingga terdiri dari sebuah himpunan state-state dan sebuah fungsi transisi dari state dengan melibatkan simbol-simbol input yang dipilih dari alfabet untuk proses transisi tersebut. Tiap satu simbol hanya mungkin mengahsilakn satu jenis state ungtuk satu state sebelumnya yang diberi simbol input tadi. Mungkin saja state yang dihasilkan adalah state itu juga dengan simbol input yang diberikan tadi tergantung dari model graph berarahnya. Penamaan automata ini didasarkan pada keterbatasan pemakaian simbol inputnya yaitu hanya satu kali untuk satu state dan hanya menghasilkan satu jenis state saja. Tapi harus diingat bahwa pada automata berhingga ini semua simbol input mutlak dipakai pada setiap state dengan mengikuti aturan tadi. Oleh karena itu harus ada pembatasan jumlah anggota himpunan simbol-simbol input, tujuannya untuk menghindari kasulitan dalam menentukan bentuk himpunan bahasa yang dapat diterima oleh suatu automata.

Definisi 4.1.1

State adalah keadaan dimana sebuahn automata berhingga bisa melakukan proses penerimaan atau penolakan suatu simbol.

Rangakaian dari state-state ini akan membentuk suatu automata bderhingga. State awalnya diberi tanda "_" dan state akhirnya diberi tanda "+". State awal dan akhir ditentukan dengan cara memilih saja dan tidak terikat oleh urutan indeks ataupun aturan tertentu.

Contoh 4.1.1

Dari contoh ini ditetapkan bahwa q_o sebagai state awal karena diberi tanda "_", sedangkan state akhirnya adalah q_2 karena telah diberi tanda "+"

Definisi 4.1.2

Simbol-simbol input adalah simbol-simbol yang diberikan pada sebuah automata berhingga untuk selanjutnya diproses melalui state-state tadi dan dituliskan dengan huruf kecil bila berupa huruf, atau bila berupa angka maka akan dituliskan dengan 0,1,2,3,...9. Notasi yang dipakai sebagai simbol input ini adalah $\sum$

Contoh 4.1.2

$\sum$ = {0,1,3} atau {a,b,c}

Definisi 4.1.3

Fungsi transisi adalah fungsi pemetaan yang memindahkan sutau state ke state lainnya setelah diberi simbol input, notasinya δ.

Definisi 4.1.4

Automata berhungga yang biasa disimbolkan A adalah pasangan dari lima tupel atas unsur-unsur pembangunnya yaitu $A = (Q, \sum, q_o, \delta, F)$ dimana

Q = Himpunan berhingga yang tidak kosong dari state-state.

$\sum$ = Himpunan berhingga yang tidak kosong dari simbol-sim,mbol input.

$q_o \in Q$ = state awal

$F \subset Q$ = Himpunan berhingga yang tidak kosong dari state akhir.

δ : Q x $\sum^*$ $\rightarrow$ 2^Q sebagai fungsi transisi. (dalam hal ini 2^Q merupakan himpunan kuasa atas Q

Contoh 4.1.4

A = ({q_o, q_1, q_2, q_3}, {1,0}, q_o, δ, {q_o}).

δ (q_o, 0) = q_2,

δ (q_o, 1) = q_1,

δ (q_1, 0) = q_3,

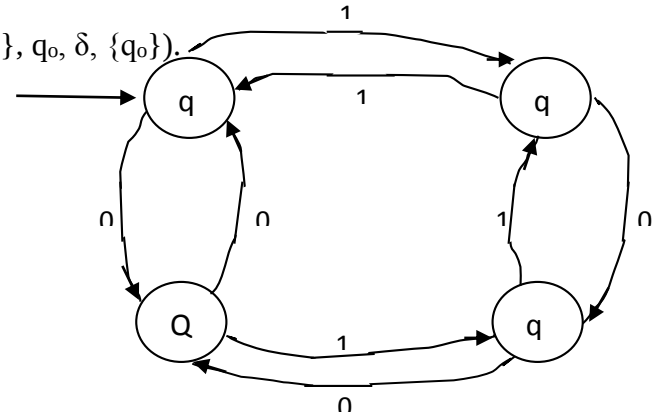
δ (q_1, 1) = q_0,

δ (q_2, 0) = q_0,

δ (q_2, 1) = q_3,

δ (q_3, 0) = q_2,

δ (q_3, 1) = q_1,



Dalam memulai suatu automata haruslah berada dalam state awal q_o selanjutnya diberikan string dan simbol dan input (elemen dari $\sum$) yang mana ditulis pada sebuah input dengan arah gerak pembacaan dari kiri ke kanan tanpa kembali.

Misal simbol pertama adalah A1 maka pada saat senin dalam state q_o, pita akan membaca a_1 dan untuk seterusnya state bergerak ke q_1; selanjutnya simbol input a_1 akan diterima jika image dari fungsi dari transisi $\delta(q_{i-1},a_1)$ terdefinisi dan tidak kosong.

Penyesuaian Bahasa oleh Automata

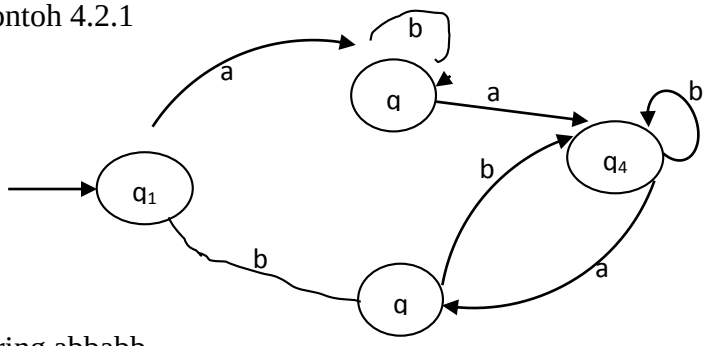
Suatu bahasa(formal), baik berupa string maupun berupa huruf akan berarti apabila susunannya benar. Bahasa seperti ini akan dikenali dan diterima serta diterjemahkan berupa output oleh suatu automata yang sesuai bentuk produksi bahasa tadi.Karena penerimaan bahasa oleh automata harus memenuhoi aturan tertentu dan tidak sembarang diterima antara suatu bahasa dengan sembarang sebalikanyapun boleh berlaku yaitu suatu automata bisa dipaksakan membuat bahasa yang sesuai.

Definisi 4.2.1

Sebuah string dari simbol-simbol input $a_1a_2a_3a_4.....a_n \in \sum^+$ akan derima oleh automata hingga jika terdapat rangkaian dari state-state $q_0,q_1,q_2,.....q_n$ sedemikian hingga $q_{i+1},= \delta(q_i,a_{i+1})$ dan $q_n \in F$

Atau dengan redaksi lain “sebuah string x akan diterima oleh automata hingga jika ada $q_0 \in Q$ sehingga $\delta(q_0,x) = p$ untuk suatu δ dan $p \in F^n$.

Contoh 4.2.1



String abbabb

$\delta(q_1,abbabb) = \delta(\delta(\delta(\delta(\delta(q_1,a),b),b),a),b),b)$

$= \delta(\delta(\delta(\delta(\delta(q_2,b),b),b),a),b),b)$

$= \delta(\delta(\delta(\delta(q_2,b),b),b),a),b),b)$

$= \delta(\delta(\delta(q_2,a),b),b)$

$= \delta(\delta(q_4,b),b)$

$= \delta(q_4,b)$

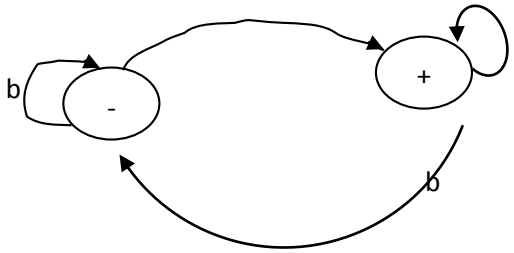
Definisi 4.2.2

Himpunan string yang diterima oleh automata hingga A disebut himpunan reguler dinotasikan dengan T (A), dimana; $T(A) = \{x \mid x \in .\sum^+ \wedge \delta(q_0,x) = q_n\}$

Pendefinisin di atas dapat diinterpretasikan sebagai berikut ;

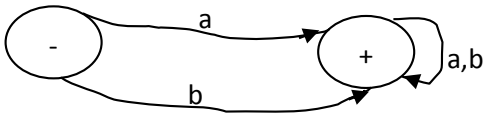
Untuk suatu string $x \in .\sum^+$, string x akan diterima oleh automata hingga A jika $\delta(q_0,x) = p$ untuk suatu δ dan $p \in F$. Himpunan dari semua string yang diterima oleh autamata hingga a disebut penerimaan bahasa oleh mesin A dan dinotasikan dengan T(A) sedemikian sehingga $T(A) = \{x \mid \delta(q_0,x) = p \in F\}$ dan bahasa yang diterima oleh automata hingga ada bahasa reguler yang dibangun oleh gramatika reguler lewat aturan produksi tertentu, makanya T (A) disebut himpunan reguler.

Contoh 4.2.2 a



Maka $T(A) = \{a \mid x \in .\sum^+, ; x = \text{string berakhiran } a\}$

Contoh 4.2.2 b



Maka T(A) = {(a+b)(a+b)*}

Automata Berhingga dengan Output.

Automata yang dibahas sebelumnya yang didefinisikan sebagi peralatan mengenal, sekaligus menerjemah suatu bahasa, belum bisa memberikan suatu output terhadap input yang diberikan. Hal ini terjadi karena belum dilengkapi dengan sistem yang dapat memberikan output bila ada input yang diberikan.

Ada dua kelompok automata berhingga beroutput yaitu kelompok *mesin moore* dan kelompok *mesin Mealy*. Kedua kelompok mesin ini mempunyai persamaan dan perbedaan dalam proses kerjanya. Mesin moore memberikan output bila ada input yang diberikan melalui state-statenya. Sedangkan mesin Mealy bila diberikan input, akan memberikan jawaban berupa output melalui transisi state-statenya. Namun kedua mesin ini mepunya persmaan yaitu keduanya memetakan input ke output.

Definisi 4.3.1

Mesin moore bisa disimbolkan dengan Mo, terdiri dari enam-tupel yaitu ($Q, \sum, \Delta, \delta, \chi, q_o$) dimana : $Q, \sum, \delta, q_o$ sama seperti pada automata berhingga. Δ = Himpunan berhingga alfabet output

$$\chi : Q \rightarrow \Delta$$

Contoh 4.3.1

Mo = ($\{q_o,q_1,q_2\}, \{1,0\}, \{0,1\}, \chi, q_o$) dimana

$\delta (q_o,0) = q_2$ $\quad\quad$ $\chi (q_0) = 0$

$\delta (q_o,1) = q_1$ $\quad\quad$ $\chi (q_1) = 1$

$\delta (q_o,1) = q_1$ $\quad\quad$ $\chi (q_1) = 1$

Mesin moore ini mempunyai jumlah nggota alfabet output yang bisa sama atau tidak dengan jumlah anggota lain. Dari alfabet itulah nantinya trbentuk suatu string output yang merupakan hasil dari string input. Oleh karen itu, panjang string sama dengan panjang string output. Pada mesin moore ini, penempatan simbol input sama seperti dengan automata hingga yang telah dibahas. Sementara penempatan simbol yang selalu siap menerjemahkan bila ada input yang masuk, harus berada tepat

disamping statenya. Tiap state harus selalu ada simbol output disampingnya yang dipisah dengan tanda garis miring(/).

Karena mesin moore mempunyai state awal maka input pertama langsung terbaca dan memberikan output. Jadi, output muncul pada saat proses tepat berada dalam satu lingkarandengan staten tertentu. Sehingga bila proses sedang beada dalam keadaan transisi, maka mesin belum bisa memberikan output meskipun perangkat lunak untuk output itu sudah ada. Semua sifat automata berhingga yang sebelumnya telah dibahas, merupakan sifat mesin moore haya saja ada penambahan fasilitas/perangkat outputnya.

Semua elemen dari alfabet output harus dipakai dan hanya satu simbol untuk satu state karena bila output tidak dipakai semua maka outputnya mungkin hanya satu macam simbol penyusunnya meskipun diberi input yang berbeda.

Satu simbol output bisa dihasilkan dari input yang berbeda. Hal ini disebabkan ketidaktunggalan rusuk yang masuk ke suatu state. Sedangkan tipe rusuk mungkin mempunya simbol input yang berbeda. Itulah sebabnya input yang berbeda kadang menghasilkan output yang sama

Contoh

String input I = 01011 outputnya 01111

String input II = 01000 outputnya 01111

Ada perbedaan kata “dipakai semua” anatars imbol input dengan simbol output. Simbol input harus “dipakai semua” dalam setiap state. Sedangkan simbol output “dipakai semua” dalam lingkup keseluruhan rangkaian, bukan hanya pada saat state tertentu (seperti contoh 4.3.1).

Berdasarkan contoh, dan bila dimisalkan ada string input 01011 maka proses pembentukan outputnya sebagai berikut :

$$
\begin{aligned}
\delta(qo, 01011) &= (q_o,0) \\
&= \delta(q_2,1), && \chi(q_2) = 0 \\
&= \delta(q_1,0), && \chi(q_1) = 1 \\
&= \delta(q_1,1), && \chi(q_1) = 1 \\
&= \delta(q_0,1), && \chi(q_0) = 1 \\
&= q_1 && \chi(q_1) = 1
\end{aligned}
$$

Dapat dilihat bahwa panjang string output sama dengan panjang string input. Semua elemen output sama dengan elemen input, hanya yang membedakannya adalah

urutannya saja, sehingga tidak tertutup kemungkinan suatu saat ada string input sama persis string output. Tergantung bagaimana penempatan simbol-simbol input dan simbol-simbol outputnya (hal ini penulis belum bisa menemukannya).

Definisi 4.3.2

Sebuah mesin mealy yang biasa disimblkan Me terdiri 6-tupel yaitu $(Q, \sum, \Delta, \delta, \chi, q_o)$ dimana : Q, $\sum$, δ, q_o sama seperti pada automata berhingga.

Δ = Himpunan berhingga alfabet output

$\chi : Qx\sum \rightarrow \Delta$

Contoh 4.3.2

Me = $(\{q_o,q_1,q_2\}, \{1,0\}, \{0,1\}, \chi, q_o)$ dimana

$\delta(q_o,0) = q_1$ $\quad$ $\chi(q_0,0) = 1$ $\quad$ $\chi(q_0,1) = 0$

$\delta(q_o,1) = q_2$ $\quad$ $\chi(q_1,0) = 1$ $\quad$ $\chi(q_1,1) = 1$

$\delta(q_1,0) = q_1$ $\quad$ $\chi(q_2,0) = 1$ $\quad$ $\chi(q_2,1) = 0$

Mesin mealy (Me) ini rangkaianya hampir sama dengan mesin moore. Hanya yang membedakan adalah penempatan simbol outputnya bukan pada statenya, melainkan pada posisi transisi dari state ke state. Oleh karena itu pada mesin mealy, output tercetak bukan saat membaca state awal melainkan saat membaca simbol input pertama. Jadi, output tercetak apabila pembacaan input sedang berlangsung diantara state statenya. Artinya, proses percetakan output berhenti bila proses sedang berada tepat pada dalam lingkaran state.

Sifat sifat lainya sama seperti mesim moore, hanya penempatan simbol outputnya yang membedakanya. Itulah sebabnya mesin mealy dikatakan memberikan output melalui proses transisi. Kelihatanya mesin mealy seakan akan tidak mempunyai state awal, ini hanya terletak pada proses pemberian outputnya saja.

3. Kesimpulan

Bahasa yang digunakan sehari-hari dapat dipakai sebagai acuan untuk menyusun bahasa formal seperti yng dipakai berkomunikasi antara komputer dengan penggunanya. Bahasa formal ini terbentuk dengan suatu aturan tertentu yang disebut gramatika. Konsep aljabar sangat relevan dengan teori bahasa formal yaitu satu simbol/kata hanya mempunyai hanya satu arti. Perbedaan jenis/tipe bahasa formal ditentukan oleh gramatika pembangunnya. Artinya bahasa formal tipe-i dibangun oleh gramatika tipe-i. Dalam hal ini hanya bahasa reguler sebagi bahasa tipe tiga yang akan

dibahas. Selanjutnya bahasa formal tadi akan dikenali oleh suatu peralatan pengenal yang disebut automata. Auotomata ini dibagi dalam beberapa jenis/tipe, tiap jenis hanya bisa mengenali satu jenis bahasa formal. Jadi, untuk bahasa (formal) reguler, automatanya adalah automata berhingga. Automata berhingga memberikan output dalam dua cara yaitu dengan melalui proses transisi dari state ke statenya dan dari state itu sendiri. Output yang dihasilkan melalui proses transisi merupakanj kelompok mesin mealy sedangkan output yang diahsilkan melaui statenya adalah kelompok mesin moore. Automata berhingga ini bisa untuk menguji layak tidaknya suatu rangkaian elektronika dipakai. Untuk tujuan ini, rusuk-rusk dalam automata dimisalkan sebagai kabel-kabel sedangkan state-statenya sebagi perangkat elektroniknya yang selaian kabel tadi seperti resistor, kapsitor. Automata berhingga dihubungkan dengan rangkaian elektronik biasa disedbut tranduser.

Daftar Pustaka

[1] Alexander Ollongren, *Definition of Programming Languages by Interpreter Automata*, Academic Press, London New York San Fransisco, 1974.
[2] John E. Hopcrff and Jeffrey D. Ullman, *Introduction to Automata Theory Languages and Computation*, Addison Wesley Publishing Company, Philippimes, 1979.
[3] J.P. Tremblay and R. Manobar, *Discrete Mathematical Strucures with Applications to Computer Science*, McGraw-Hill Book Company, New York.
[4] Sankar K. Pal and Dwijesh K., Dutta Majumder, *Fuzzy (pendekatan Matematik untuk pengenalan pola)*, Oxford University Press.
[5] Sarwah, *Analisis Deskripsi Bahasa Yang Bersesuaian Dengan Model Automata*, Universitas Hasanuddin, 1998.